Recurrent competitive networks can learn locally excitatory topologies

Jug, Florian ; Cook, Matthew ; Steger, Angelika

Abstract: A common form of neural network consists of spatially arranged neurons, with weighted connections between the units providing both local excitation and long-range or global inhibition. Such networks, known as soft-winner-take-all networks or lateral-inhibition type neural fields, have been shown to exhibit desirable information-processing properties including balancing the influence of compatible inputs, deciding between incompatible inputs, signal restoration from noisy, weak, or overly strong input, and the ability to be used as trainable building blocks in larger networks. However, the local excitatory connections in such a network are typically hand-wired based on a fixed spatial arrangement which is chosen using prior knowledge of the dimensionality of the data to be learned by such a network, and neuroanatomical evidence is stubbornly inconsistent with these wiring schemes. Here we present a learning rule that allows networks with completely random internal connectivity to learn the weighted connections necessary for implementing the "local" excitation used by these networks, where the locality is with respect to the inherent topology of the input received by the network, rather than being based on an arbitrarily prescribed spatial arrangement of the cells in the network. We use the Siegert approximation to leaky integrate-and-fire neurons, obtaining networks with consistently sparse activity, to which we apply standard Hebbian learning with weight normalization, plus homeostatic activity regulation to ensure full network utilization. Our results show that such networks learn appropriate excitatory connections from the input, and do not require these connections to be hand-wired with a fixed topology as they traditionally have been for decades.

DOI: https://doi.org/10.1109/IJCNN.2012.6252786

Recurrent Competitive Networks Can Learn Locally Excitatory Topologies

Florian Jug
Institute of Theoretical Computer Science
Department of Informatics
ETH Zurich
Zurich, Switzerland

Matthew Cook
Institute of Neuroinformatics
University of Zurich
and ETH Zurich
Zurich, Switzerland

Angelika Steger
Institute of Theoretical Computer Science
Department of Informatics
ETH Zurich
Zurich, Switzerland

***Abstract*—A common form of neural network consists of spatially arranged neurons, with weighted connections between the units providing both local excitation and long-range or global inhibition. Such networks, known as soft-winner-take-all networks or lateral-inhibition type neural fields, have been shown to exhibit desirable information-processing properties including balancing the influence of compatible inputs, deciding between incompatible inputs, signal restoration from noisy, weak, or overly strong input, and the ability to be used as trainable building blocks in larger networks. However, the local excitatory connections in such a network are typically hand-wired based on a fixed spatial arrangement which is chosen using prior knowledge of the dimensionality of the data to be learned by such a network, and neuroanatomical evidence is stubbornly inconsistent with these wiring schemes. Here we present a learning rule that allows networks with completely random internal connectivity to learn the weighted connections necessary for implementing the "local" excitation used by these networks, where the locality is with respect to the inherent topology of the input received by the network, rather than being based on an arbitrarily prescribed spatial arrangement of the cells in the network. We use the Siegert approximation to leaky integrate-and-fire neurons, obtaining networks with consistently sparse activity, to which we apply standard Hebbian learning with weight normalization, plus homeostatic activity regulation to ensure full network utilization. Our results show that such networks learn appropriate excitatory connections from the input, and do not require these connections to be hand-wired with a fixed topology as they traditionally have been for decades.**

I. Introduction

A common form of neural network, often referred to as a soft-winner-take-all (WTA) network [1] or lateral-inhibition type neural fields [2], consists of spatially arranged units, with strong mutual excitation between close neighbors, fading to mild inhibition between well-separated units.

This structure leads to localized activity, where a local group of active neurons generates mutually excitatory input to sustain the activity, while the rest of the network has its activity suppressed by the long-range inhibition.

If there are two or more regions of localized activity, then the activity in each of these regions inhibits the other regions, with the largest amount of inhibition arriving to the most weakly active region and the least amount of inhibition reaching the most strongly active region. This leads to the weakly active regions becoming even more suppressed. In the absence of sustained external input, the weaker regions will become completely inactive, leaving a single "winning" region of strong activity. These dynamics are the source of the name "soft-winner-take-all" [1]. The term "soft" is used to distinguish the dynamics from those of a "hard" winner-take-all circuit in which a single unit, rather than a local group with soft boundaries, is the winner.

Self-organizing maps, to name a famous, classical example, use a very similar mechanism to pick the units that are subject to the associated learning rule [3]. But WTA networks have many uses in neural information processing. For example, given excitatory input from two different sources, with the contribution from each source concentrated in a localized region of the network. If the two regions are close to each other, the result will be a region of activity at an intermediate position which compromises between the two inputs. This is sometimes referred to as "cue integration" [4], [5], [6] in analogy with how we merge multiple imprecise cues to form improved estimates.

If, on the other hand, the two input-receiving regions are not close to each other, so there is no compromise value which is largely consistent with both inputs, then the network makes a decision between these inputs, focusing its activity around the input with stronger support (higher input rates), while significantly reducing or eliminating activity around the input with weaker support [1], [2], [4]. This is a simple form of decision making.

Furthermore, the stereotyped responses generated by these networks makes them robust to situations where the input is very noisy, or weaker or stronger than usual. Another way of viewing this, if we think of the network as a filter which turns its input into its output, is that the network is performing signal restoration on signals whose structure should consist of a stereotypically-shaped localized region of activity [1].

In addition to these features, these types of soft-winner-take-all structures can be combined in a modular way to form

larger networks which are able to learn and use underlying relationships existing in data presented to the network [6].

Unfortunately, neocortical neuroanatomy does not support the idea of localized excitation combined with global inhibition. In fact, nearly the opposite is true: All long-distance connections are excitatory. Both excitatory and inhibitory neurons target the immediate local neuropil, but the inhibitory neurons tend to have a slightly smaller axonal arborization, and secondary arborizations beyond the local region are nearly always from excitatory cells. Of course, this could still be consistent with long-range inhibition, if for example the long-range excitatory connections tended to prefer to target inhibitory cells, but there does not appear to be any evidence for this [7]. Indeed, much synaptic connectivity appears to indiscriminately target whatever dendrites are in the area [8], [7], [9], [10].

It is worth pointing out that the "localized region" of activity occurring in soft-winner-take-all networks might not correspond to a compact region of neuropil, but rather be with respect to a more abstract "feature space", where a region is defined as a set of neurons which are responsive to a similar set of features, regardless of their precise cranial position. The coordinates of a point in feature space correspond to values of the various measurable features which are represented in that space. For example, primary visual cortex has neurons which are tuned to a combination of orientation, direction of motion, spatial frequency, ocular dominance, and retinal location [11]. The corresponding feature space has these as dimensions, and each neuron which is tuned in all of these dimensions can be assigned a location (indeed a receptive field) in this high-dimensional feature space. A localized region of activity in the feature space might appear as a disordered group of active neurons when viewed in terms of their position in the neuropil.

However, this apparent disarray seems unlikely to explain the discrepancy between the wiring needed for these networks and the wiring seen in neocortex. While WTA networks use relatively few strong excitatory connections along with a large number of weakly inhibitory connections, the majority of cortical synapses are excitatory with inhibitory synapses being strong and effective [12], [13]. In short, neuroanatomical knowledge suggests that the wiring schemes used in classical soft-winner-take-all networks are simply unrealistic.

In this paper we show how simple learning rules can yield network connectivities equivalent to the local excitatory connections used in soft-winner-take-all networks. We use small networks with only a few hundred neurons, corresponding to cortical subnetworks such as the layer 2/3 subpopulation of a cortical minicolumn. We use excitatory and inhibitory cells in the same proportions as found in the cortex, namely a ratio of four times more excitatory than inhibitory cells. The inhibitory cells provide for global inhibition throughout the population, in proportion to the current activity (minus a constant), thus regulating the total amount of excitatory activity so that it stays roughly constant. This effectively implements a form of divisive normalization [14].

To study learning mechanisms, long periods of time must be simulated, which is computationally very expensive for spiking network simulations. To speed up the development cycle, we use the Siegert approximation to leaky integrate-and-fire neurons, giving us a mean firing rate model which is closely tied to the spiking model, for example using the same set of parameters.

Hebbian learning is used on the recurrent excitatory weights, so units that respond at the same time become mutually excitatory, and weight normalization keeps this excitation bounded while also making sure that units with anti-correlated activity have their mutual weights reduced. This leads to the formation of mutually excitatory connections between units that respond to similar inputs. This is exactly the type of local excitation that has traditionally been wired by hand into these types of networks.

We also use a homeostatic activity regulation mechanism, in the style of synaptic scaling [15]. This applies an activity-based bias, so that neurons with low average activity get their weights increased, becoming more likely to be active, while units with high average activity are biased towards a lower activity level. The point is to avoid "dead" units by ensuring that all units participate in the network's output at least some of the time. Once a previously-inactive unit starts participating in the output, it will start strengthening its connections to other units which are active at the same time, via the Hebbian learning. The Hebbian learning with weight normalization effectively causes different local regions (i.e. groups of neurons that are active together) to compete for getting the previously-inactive unit to join their group.

Once all of the units have started to participate in the network output, then the homeostatic activity regulation will spread them out so they are each active roughly the same fraction of the time. This ensures an even activity level in the response from the network independent of the content of the response.

The local excitatory connectivity learned by our networks is local with respect to the feature space or input space seen by the network, the space from which sample inputs are drawn.

If our network is exposed to a one-dimensional set of stimuli, then it develops a recurrent weight matrix whose strong excitatory weights are local with respect to this one-dimensional space.

If the network is exposed to a two-dimensional set of stimuli, then it develops recurrent excitatory weights which are local with respect to this two-dimensional space.

In general, the local excitatory connections learn to mimic the topology of the input space. This presents another advantage of our networks over traditional hand-wired networks: Whereas traditionally the network designer has needed to know ahead of time what sort of data will be seen by the network, so that the appropriate topology can be hand-wired into the network, our networks figure out on their own at run-time what the appropriate structure of recurrent connectivity is for the observed inputs.

II. Neuron Model

We use the Siegert mean-firing-rate approximation [16], [17], [18] to a leaky integrate-and-fire neuron, as shown in Figure 1. This approximation is based on the assumption that the inputs are uncorrelated Poisson spike trains. For ongoing activity in a recurrent network, this will usually not be the case, leaving the question of how reasonable the approximation is.

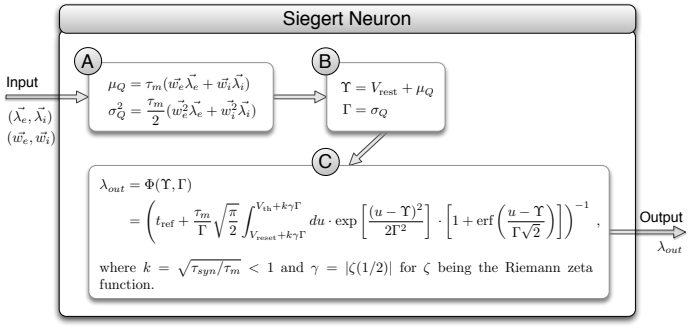


Fig. 1: The Siegert Node. The diagram shows the necessary steps for a Siegert neuron [16], [17], [18] to compute its output rate λ_{out}. It receives excitatory and inhibitory inputs where $\vec{\lambda_e}$ ($\vec{\lambda_i}$) are incoming excitatory (inhibitory) rates and $\vec{w_e}$ ($\vec{w_i}$) are the corresponding synaptic weights. The formulas in (A), (B), and (C) are sequentially evaluated, where τm, V_{rest}, V_{reset}, V_{th}, and t_{ref} are the membrane leakage, the resting potential, the reset potential, the threshold potential, and the absolute refractory time, respectively. Note that these are precisely the same parameters used for leaky integrate-and-fire (LIF) neurons [19]. Note that $\vec{v^2}$ denotes the element-wise square $(v_1^2, v_2^2, \ldots, v_{|\vec{v}|}^2)^T$, and $\vec{w}\vec{\lambda}$ is the vector dot product of $\vec{w}$ and $\vec{\lambda}$.

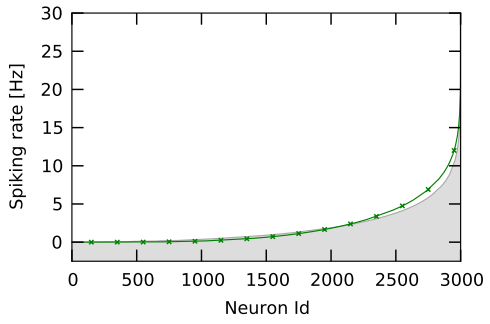


Fig. 2: Activation of the excitatory subpopulation of a leaky integrate-and-fire network and an equivalent Siegert Recurrent Competitive Network. The network is described in Section III and in [13], [12]. Note that the abscissa is not continuous but simply plots the activation of all 3000 neurons of the simulated recurrent network. Also note that these neurons where sorted by ascending activity values. The gray area shows these sorted activities of the Recurrent Competitive Network simulated using leaky integrate-and-fire neurons. In green we have plotted the sorted activation profile of an equivalent Recurrent Competitive Network evaluated using the Siegert approximation. The activation profile turns out to be similar for a wide range of possible inputs.

We tested this approximation by comparing the network activity profile between spiking simulations and Siegert-based simulations. The result is as shown in Figure 2. Similar results were obtained under many different parameter settings.

The parameters we used, for excitatory and inhibitory units, are given in Table I.

TABLE I: Neural and synaptic parameters for both the leaky integrate-and-fire and the Siegert simulations. Parameters τm, V_{rest}, V_{reset}, V_{th}, and t_{ref} are the membrane leakage, the resting potential, the reset potential, the threshold potential, and the absolute refractory time, respectively. AMPA (*Alpha-Amino-3-Hydroxy-5-Methyl-4-Isoxazole Propionic Acid*) and NMDA (*N-Methyl-D-Aspartat*) are excitatory synaptic currents parametrized by τ's denoting the rise and decay times of the respective current. The values for ρ_{AMPA} give the ratio of AMPA and NMDA receptors in the postsynaptic process of the respective neuron. GABA (*Gamma-Aminobutyric Acid*) is an inhibitory synaptic current that is, like the AMPA current, modeled by an exponential decay.

Excitatory Cells

parameter	default	unit
V_{rest}	-65.0	mV
V_{reset}	-65.0	mV
V_{th}	-52.0	mV
τm	20.0	–
t_{ref}	2.0	ms
ρ_{AMPA}	0.5	–
τ_{AMPA}	1.5	ms
$\tau_{\text{NMDA,rise}}$	10.0	ms
τ_{NMDA}	100.0	ms
τ_{GABA}	5.5	ms

Inhibitory Cells

parameter	default	unit
V_{rest}	-60.0	mV
V_{reset}	-60.0	mV
V_{th}	-40.0	mV
τm	10.0	–
t_{ref}	1.0	ms
ρ_{AMPA}	1.0	–
τ_{AMPA}	1.5	ms
τ_{GABA}	5.5	ms

III. Network Structure

The structure of the network is as shown in Figure 3.

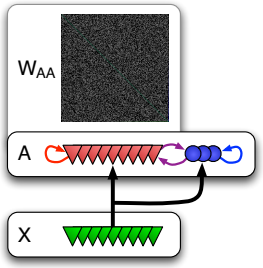


Fig. 3: Recurrent Competitive Network structure. The input layer (X) shown in green feeds input to the Recurrent Competitive Network (A) which consists of both excitatory cells (red) and inhibitory cells (blue), all recurrently connected with parameters given in Table II. The sparse matrix (W_{AA}) shows the initial connection weights of the recurrent connections between the pyramidal cells of A (red arrow). These are the weights that will be affected by learning.

The activation level of each unit in the input population is fixed for a given input presentation. The input population is connected one-to-one to the main layer above it. Thus each input unit's activity level simply represents a Poisson rate to be received by the target unit in the main layer.

The Recurrent Competitive Network (layer A in Figure 3) has its dynamics modeled using the Siegert neuron described above. The connections between subpopulations, shown with arrows in the figure, are sparse (see Table II) and randomly initialized. Only the recurrent weights between excitatory neurons (the red arrow in the figure) undergo learning; other weights do not change.

TABLE II: The set of parameters for an Recurrent Competitive Network setup containing 256 input-, 256 excitatory- and 64 inhibitory neurons.

Network parameters							
n_i, n_p	n_b	p_{ip}	p_{ib}	p_{bp}	p_{pb}	p_{pp}	p_{bb}
		–	–	0.25	0.25	0.50	0.50
256	64	w_{ip}	w_{ib}	w_{bp}	w_{pb}	w_{pp}	w_{bb}
		1.25	3.00	−2.00	3.00	1.25	−2.00

IV. Learning Rule

The function **learnWeights**($network$,W) listed below describes the learning rule in pseudo-code. The weight modification happens in line 6. The expressions c.pre and c.post denote the presynaptic and postsynaptic neuron, respectively.

Function **learnWeights** ($network$,W):

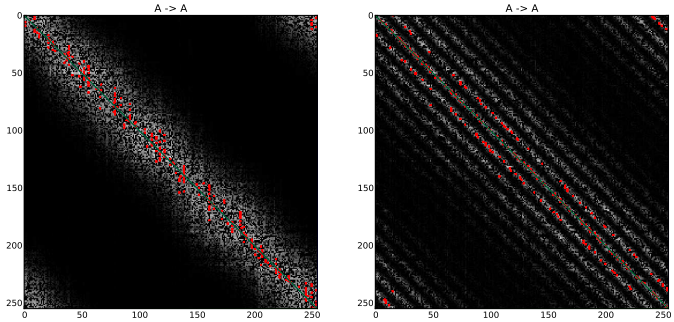
1: **for each** n **in** $network$.neurons **do**
2: $oldSum$ = $newSum$ = 0
3: /* compute new weights */
4: **for each** c **in** $n.W$ **do**
5: $oldSum$ += c.weight
6: c.weight = c.weight $+ \alpha \cdot (c.\text{pre.rate} \cdot c.\text{post.rate})^k$
7: $newSum$ += c.weight
8: **end for**
9: /* normalize weights */
10: **for each** c **in** $n.W$ **do**
11: c.weight = c.weight $\cdot \frac{oldSum}{newSum}$
12: **end for**
13: **end for**

With the constant k used in line 6, we can control whether the weight changes are linear ($k = 1$), superlinear ($k > 1$), or supralinear ($k < 1$) with the product of c.pre.rate and c.post.rate, the rates of the presynaptic and postsynaptic neuron, respectively. For $k = 1$ this rule is a classical Hebbian learning rule with learning rate α. In the simulations performed for creating the shown figures we used the values $\alpha = 0.04$ and $k = 2$. Using $k = 1$ leads to very similar simulation results where the strong weights along the diagonal of Figure 4 would be a bit wider (data not shown).

It turns out that in cases where the recurrent excitatory connections W_{AA} are relatively strong, learned weight matrices have the tendency to become block diagonal matrices, which leads to dynamics where the system settles into one of several discrete states. We use weaker recurrent weights, leading to smooth fuzzy diagonal matrices with continuous dynamics rather than such discrete behavior.

Another factor affecting whether a continuous (smooth) or discrete (blocky) weight matrix is obtained is whether the learning is done when a new input is first presented or only after the network has converged. Waiting for convergence has the effect of emphasizing the influence of the existing recurrent connections, while learning at stimulus onset gives greater emphasis to correlations present in the input. Performing learning at stimulus onset is motivated by biophysiology. Experimental evidence suggests that backpropagating action-potentials (BAPs) are necessary for long term potentiation (LTP) [20]. The propagation of these BAPs are based on a chain reaction that is much less reliable than the one used for action potential propagations in axons. Inhibitory currents (GABA currents) in dendritic branches can disrupt this chain reaction [21], which in turn renders the synapses along subsequent branches unable to perform further LTP. This effect grows with GABA currents transmitted between the cells in a neural population. These are initially low, increasing after stimulus onset.

With the constant k used in line 6, we can control whether the weight changes are linear ($k = 1$), superlinear ($k > 1$), or supralinear ($k < 1$) with the product of c.pre.rate and c.post.rate, the rates of the presynaptic and postsynaptic neuron, respectively. For $k = 1$ this rule is a classical Hebbian learning rule with learning rate α. For the simulations in this paper we used the values $\alpha = 0.04$ and $k = 2$. Using $k = 1$ leads to very similar simulation results where the strong weights along the diagonal of Figure 4 would be a bit wider. The use of a value $k > 1$ is motivated by neurophysiological findings (see e.g. [22]).



(a) **(b)**

Fig. 4: Learned weights W_{AA}. Each plot shows the learned recurrent weights among the pyramidal cells of a Recurrent Competitive Network. Brightness indicates weight strength. Red markers show the largest value in each row. **(a)** Weight matrix after training on 1D-inputs. **(b)** Weight matrix after training on 2D-inputs. In each case, training converged after a few hundred input presentations.

V. Homeostatic Activity Regulation

It is known that neurons actively regulate their spiking activity in order to maintain a certain activity level or activity range [15], [23], [24].

In order to include this type of homeostatic activity regulation in our simulations we model a mechanism known as *Synaptic Scaling*. The basic idea is that a neuron which is too active (to little active), will scale down (up) all its incoming synaptic connection weights by the same factor f_{HAR}. This causes the neuron to adjust the expected postsynaptic currents and therefore to adjust its overall activity.

For convenience we actually do not change the individual synaptic weights in our simulations. Instead we give each neuron n a state variable n.har that is initialy set to be 1.0. Before using a synaptic weight w we compute the effective weight $w_{\text{eff}} = w \cdot n$.har, which is equivalent to the process described above. This allows the weight normalization step of the Hebbian learning to be independent of the homeostatic activity regulation.

VI. Input Patterns

The two kinds of input shown in Figure 5 are fed into identical networks in our simulations. The network parameters are the same in all cases, as given in Tables I and II. The only difference between simulations is in the presented input.

All inputs we use wrap around in order to avoid border effects. In other words, the input topology is a ring in the one-dimensional case and a torus in the two-dimensional case. For better readability, the mathematical description given below ignores this wrap-around nature.

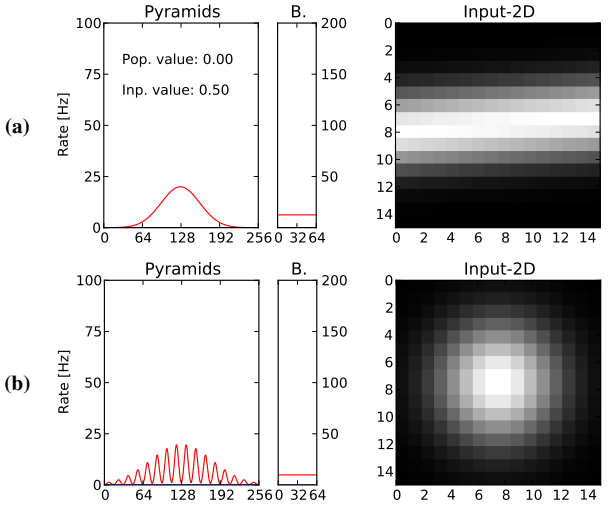


Fig. 5: The two types of input. The left sides show the input rates fed from the $n_i = 256$ input neurons to the pyramidal and basket cell population. The right sides show the same data by color encoding the input rate (white being highest rate, black being zero). In addition the 256 values have been rearranged in a 2-dimensional grid of size 16^2. **(a)** A one-dimensional (1D) population-coded input pattern centered at $x = 0.5$. **(b)** A two-dimensional (2D) population-coded input pattern centered at $\mathbf{v} = (0.5, 0.5)$.

A. One-Dimensional Input (1D)

The input population encodes a single scalar value $x \in [0, 1]$ using a population code.

There are n_p groups of input neurons. Each group G_g, $g \in \{1, \ldots, n_p\}$, consists of n_i individual cells. Every cell $c_{g,i} \in G_g$ has a preferred input stimulus value $x_{\text{pref}}(g) = \frac{g}{n_p}$ assigned to it.

We use a Gaussian function $f_{\sigma, r_{\max}}(x)$ as population activation pattern:

$$f_{\sigma, r_{\max}}(x) = r_{\max} \cdot e^{-\left(\frac{x}{2 \cdot \sigma^2}\right)} \tag{1}$$

The peak of f is at $r_{\max}$, which corresponds to the maximum input frequency in Hz.

For input x, each input cell's output rate is then given by:

$$r(c_{g,i}, x) = f_{\sigma, r_{\max}}(x_{\text{pref}}(g) - x), \tag{2}$$

the value of f shifted to x, at the cells preferred value $x_{\text{pref}}(g)$. Figure 5(a) shows an example of a 1D-input for $x = 0.5$.

We choose a random input pattern by drawing a random value x uniformly at random from the unit interval. All results shown below are done using the parameters $\sigma = \frac{n_p}{8} = 32$ and $r_{\max} = 40$ Hz.

B. Two-Dimensional Input (2D)

This type of input encodes two independent scalar values $x \in [0, 1]$ and $y \in [0, 1]$.

To each of the n_p input neuron groups G_g, $g \in \{1, \ldots, n_p\}$, n_p being a square number, we assign a vector:

$$\begin{aligned} \mathbf{v}_{\text{pref}}(g) &= \begin{pmatrix} x_{\text{pref}}(g) \\ y_{\text{pref}}(g) \end{pmatrix}, \text{with} \\ x_{\text{pref}}(g) &= \frac{g \bmod \sqrt{n_p}}{\sqrt{n_p}}, \text{and} \\ y_{\text{pref}}(g) &= \frac{\lfloor g / \sqrt{n_p} \rfloor}{\sqrt{n_p}}. \end{aligned} \tag{3}$$

This assigns the n_p input groups to the points in a two-dimensional regular grid spanning the unit square.

Since we have a two-dimensional input space, we use a two-dimensional Gaussian function:

$$f^2_{\sigma, r_{\max}}(\mathbf{v}) = r_{\max} \cdot e^{-\left(\frac{\sqrt{x^2+y^2}}{2 \cdot \sigma^2}\right)}, \tag{4}$$

with $\mathbf{v} = (x, y)^T$ as population activation pattern.

Each input cell's output rate is then given by:

$$r(c_{g,i}, \mathbf{v}) = f^2_{\sigma, r_{\max}}(\mathbf{v}_{\text{pref},g} - \mathbf{v}) \tag{5}$$

the value of f^2, shifted by $\mathbf{v}$, at the cells preferred value $\mathbf{v}_{\text{pref}}(i)$. Figure 5(b) shows an example of a 2D-input for $\mathbf{v} = (0.5, 0.5)^T$.

A random input pattern will be created by drawing a random vector $\mathbf{v}$ uniformly at random from the unit square. All results shown below are done using the parameters $\sigma = \frac{\sqrt{n_p}}{5} = 3.2$ and $r_{\max} = 40$ Hz.

VII. Results

The previous sections have set the stage for our goal of learning the topology of the input space by changing the weights between excitatory units in a Recurrent Competitive Network.

Here we will first have a look at the weights that result from the learning procedure. We will then show that the dynamics of such a trained Recurrent Competitive Network change significantly due to the learning process, becoming very close to the dynamics of soft-winner-take-all networks.

A. Topology of Learned Weights

We use the exact same Recurrent Competitive Network for each type of input described in Section VI. After being trained on a few hundred input examples, the recurrent weights among the excitatory cells converge to the weights shown in Figure 4. Extending the simulation for several thousand additional iterations shows that the converged state is stable.

Although the weight matrix does, of course, contain all the data we are interested in, it is not easy to interpret the result for two-dimensional inputs shown in Figure 4(b). Each column c of this matrix contains all incoming weights coming from other pyramidal cells. For two-dimensional inputs it is easier to interpret these incoming weights if they are shown not in a column but in a small two-dimensional grid. This kind of re-grouping of weights W_{AA} is shown in Figure 6.

It can be seen that for both types of input the strongest weights in the learned weight matrices are coming from cells with similar preferred input tuning (places that are close to each other with respect to the input topology).

It is important to step back and acknowledge the fact that the learned weights reflect the topology of the used input, even though the Recurrent Competitive Network only contains randomly drawn connections. Hence the network's initial connectivity and the topology of the input are very different in nature.

These results show that random networks, structurally similar to networks thought to exist in local cortical structures, can learn to deal with diverse input topologies even when their structure is fundamentally different from the network's inherent "tabula rasa" topology [9], [10].

B. Using the Learned Topologies

Here we show how trained Recurrent Competitive Networks change their dynamics in comparison to untrained networks. The main difference is that the excitation in trained networks is operating not only on the level of single cells, but on a pattern level. This means that the dynamics of the trained Recurrent Competitive Network include fusion or competition between simultaneously presented inputs, as well as denoising of inputs and completion of partial patterns.

Figures 7-9 show the network activity in an untrained as well as in a trained one-dimensional Recurrent Competitive Network when two inputs are fed simultaneously. The network fuses the inputs if they are similar enough, as in Figures 7 and 8). In cases where one of the two inputs is weaker than the other, the network strongly emphasizes this difference, as seen in Figure 9. These dynamics are very similar to the competition mechanisms proposed to exist in biological neural circuits [1] and also has striking similarities to the dynamics in WTAs [2].

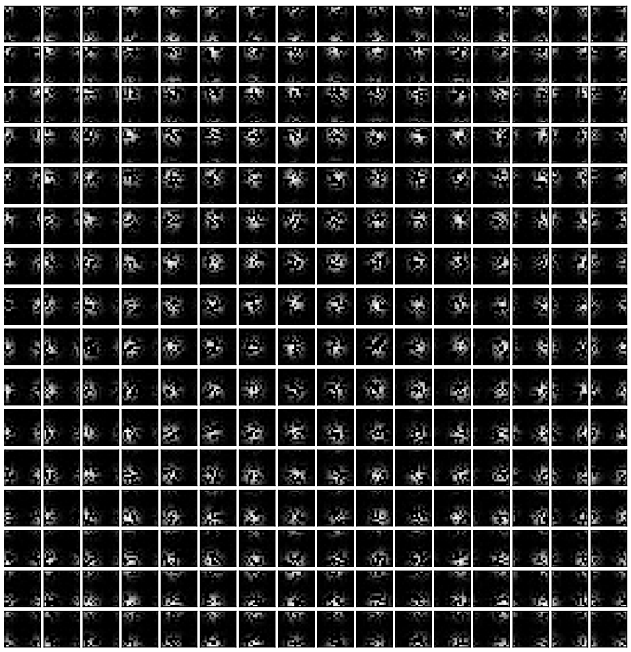


Fig. 6: Weights learned by feeding two-dimensional input as described in Section VI. The weights shown in one tile correspond to the weights in one row of the weight matrix shown in Figure 4(b). We can see that the strong inputs to a neuron (white clusters inside the tiles) come from its neighbors: the relative position of the tile in the array of tiles coincides with the position of the strongest weights inside the tile. This is the four-dimensional analogue of the smooth diagonal matrix shown in Figure 4(a).

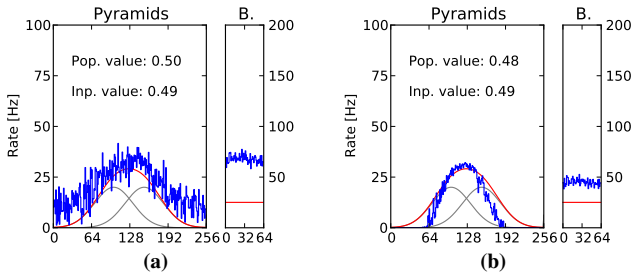


Fig. 7: Responses of two Recurrent Competitive Networks that were **(a)** untrained, and **(b)** trained using a sequence of randomly placed one-dimensional inputs. Subfigures **(a,b)** are both generated by simultaneously feeding two inputs (gray curves) to the network. In **(a)** the untrained network responds by roughly reproducing the shape of the input pattern, while in **(b)** we can see that the trained network activity is almost as sharp as the response to a single input would be.

Figures 10-11 show similar results to Figures 7-9, except that the Recurrent Competitive Network was exposed to two-dimensional inputs. In addition to fusion and competition for

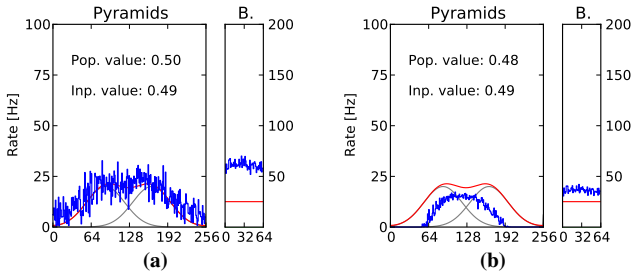


Fig. 8: Responses of two Recurrent Competitive Networks that were **(a)** untrained, and **(b)** trained, as in Figure 7. In contrast to Figure 7, the inputs (gray curves) encode more distant values, and their sum is now bimodal. In **(a)** the untrained network responds by roughly reproducing the bimodal input pattern, while in **(b)** the network activity becomes unimodal, expressing a single peak of activity at roughly the center between the peaks of the individual inputs. The trained network has fused the input signals.

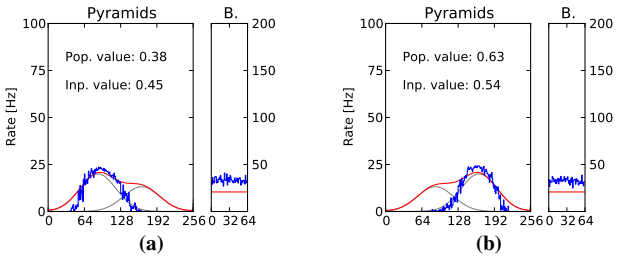


Fig. 9: Responses of a single Recurrent Competitive Network that was trained as in Figures 7(b) and 8(b). In contrast to those figures, the inputs (gray curves) now encode values using different amplitudes. The stronger input effectively competes with and suppresses the weaker input, similar to winner selection in a soft-winner-take-all network.

distinct inputs, trained Recurrent Competitive Networks are able to complete partial patterns, as shown in Figure 12. Pattern completion requires an understanding of the structure of valid patterns, which is defined by the patterns usually seen by the network. After training, the network has strengthened excitatory connections between units whose activity was highly correlated during training. This provides a secondary excitatory pathway to missing pattern parts, via excitatory connections from the extant part of the pattern.

VIII. Discussion

We have presented a network which learns to behave like a soft-winner-take-all network. The network starts with completely randomized connections with no internal topology or sense of locality upon which local excitatory connections could be based. Over time, the recurrent excitatory connections within the network are trained via Hebbian learning with weight normalization, combined with a homeostatic activity regulation mechanism along the lines of synaptic scaling. The result of this training is that the recurrent excitatory weights among the units of the network take the form of local excitatory connections, with the sense of locality coming from the topology of the input space. Thus the topology or dimensionality of

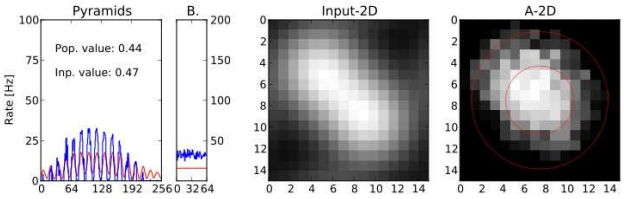


Fig. 10: Response of a Recurrent Competitive Network that was trained using a sequence of randomly placed two-dimensional inputs. The left part shows the network activity as in previous figures, while the input (red) and output (blue) are shown again in the center and right parts respectively, displayed as in Figure 5. In this case, the two inputs encode the values $\mathbf{v}_A = (0.35, 0.35)$ and $\mathbf{v}_B = (0.65, 0.65)$. The network fuses the two inputs, treating them like one larger input encoding a value between the individual inputs, at roughly $\mathbf{v}_{A+B} = (0.5, 0.5)$, circled in red in the rightmost part.

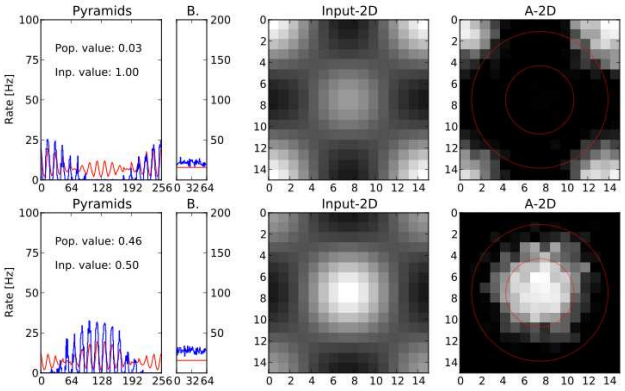


Fig. 11: Response of a Recurrent Competitive Network that was trained using a sequence of randomly placed two-dimensional inputs. Two different input configurations are shown, one in the upper row and one in the lower row. Each row is displayed as in Figure 10. In this case, the two inputs encode the values $\mathbf{v}_A = (0.0, 0.0)$ and $\mathbf{v}_B = (0.5, 0.5)$, but with different amplitudes. The upper panels show the case where $\mathbf{v}_A$ is stronger than $\mathbf{v}_B$, while the lower panels show the reverse. The stronger input effectively competes with and suppresses the weaker input, similar to Figure 9 and to winner selection in a soft-winner-take-all network.

the input space does not need to be known ahead of time, but can be left for the network to discover. Combined with the self-regulation provided through the inhibitory units, the final dynamics of the network correspond almost perfectly to traditional hand-wired soft-winner-take-all network dynamics.

Two of the three components of sharp learning [25], namely Hebbian learning and homeostatic activity regulation, are built into our network. The third component is exactly the local excitatory connectivity that arises naturally in the network we have presented here. Thus, all the components of sharp learning are present, even though only two of them (Hebbian learning and homeostatic activity regulation) were explicitly designed into the system. Sharp learning has been shown to allow consistent propagation of information in developing cortical systems [26], [12].

The automatic learning of the topology of the input space also has clear benefits when building larger networks out of

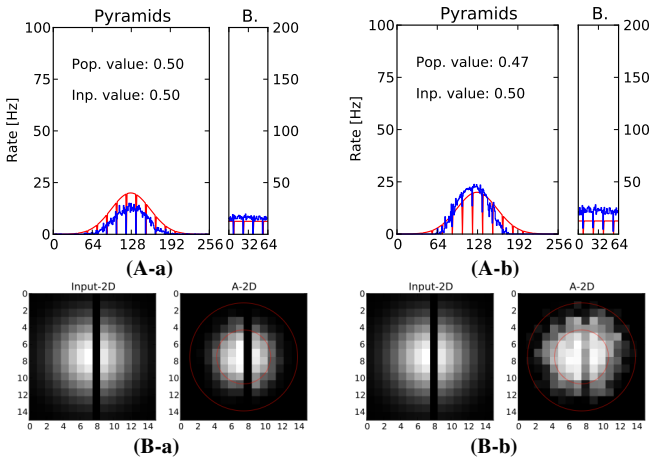


Fig. 12: Signal restoration capabilities of trained and untrained Recurrent Competitive Networks. **(A-a,A-b)** show the response to partial one-dimensional input for untrained and trained Recurrent Competitive Networks respectively. **(B-a,B-b)** show the response to partial two-dimensional input for untrained and trained Recurrent Competitive Networks respectively. Although the input pattern misses several inputs, the settled network activity is not dropping to zero at these locations.

modules of the form of our network. In this case, even the network designer may not know ahead of time what sort of data will be seen by a module deep in the network's interior. Having a module that can adapt to whatever input topology it is confronted with could be very helpful in such constructions.

Our network was designed to be biologically plausible in several ways. The ratio of excitatory to inhibitory cells matches neocortical ratios [7]. The leaky integrate-and-fire model uses parameters that are taken from experimental studies [13], [12], and our Siegert model uses these same parameters. We start with random connectivity, which is consistent with known anatomy [8], [9], [10], and use learning methods which are local and plausible. The types of input we present to our network are like the population codes which have been observed in many brain areas [4]. Not only the design of the network, but also its resulting properties are biologically plausible [1]. The types of behaviors exhibited by our trained network correspond to behaviors that are typical in biological systems, such as cue integration and decision making [4], [5], [6]. Finally, the ability of the network to adapt to an arbitrary input topology means the network is ready to be used for diverse applications, just as the same cortical microcircuit appears to be used with minor variations [27], [28] for tasks ranging from visual and auditory processing to motor control and high level reasoning.

Acknowledgments

The authors thank Johannes Lengler, Christoph Krautz, and Rodney Douglas for helpful discussions. This work was supported by funding from ETH Research Grant ETH-23 08-1 and EU Project Grant FET-IP-216593.

References

[1] R. Douglas and K. Martin, "Recurrent neuronal circuits in the neocortex," *Current Biology*, vol. 17, no. 13, pp. 496–500, 2007.

[2] S. Amari, "Dynamics of pattern formation in lateral-inhibition type neural fields," *Biological Cybernetics*, Jan. 1977.

[3] T. Kohonen, "Self-organized formation of topologically correct feature maps," *Biological Cybernetics*, Jan. 1982.

[4] S. Deneve, P. Latham, and A. Pouget, "Efficient computation and cue integration with noisy population codes," *Nature Neuroscience*, vol. 4, no. 8, pp. 826–831, 2001.

[5] W. Ma, J. Beck, P. Latham, and A. Pouget, "Bayesian inference with probabilistic population codes," *Nature Neuroscience*, Jan. 2006.

[6] M. Cook, F. Jug, and C. Krautz, "Unsupervised learning of relations," *Artificial Neural Networks–ICANN 2010*, 2010.

[7] T. Binzegger, R. Douglas, and K. Martin, "A Quantitative Map of the Circuit of Cat Primary Visual Cortex," *Journal of Neuroscience*, Jan. 2004.

[8] V. Braitenberg and A. Schütz, "Peters' Rule and White's Exceptions," *Anatomy of the cortex*, pp. 109–111, 1991.

[9] N. Kalisman, G. Silberberg, and H. Markram, "The neocortical microcircuit as a tabula rasa," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 102, no. 3, pp. 880–885, Jan. 2005.

[10] R. Perin, T. K. Berger, and H. Markram, "A synaptic organizing principle for cortical neuronal groups." *PNAS*, vol. 108, no. 13, pp. 5419–5424, Mar. 2011.

[11] N. V. Swindale, "How many maps are there in visual cortex?" *Cerebral Cortex*, vol. 10, no. 7, pp. 633–643, July 2000. [Online]. Available: http://cercor.oxfordjournals.org/cgi/content/abstract/10/7/633

[12] C. Krautz, "On principles of cortical computation," Ph.D. dissertation, ETH Zurich, 2012.

[13] F. Jug, "On competition and learning in cortical structures," Ph.D. dissertation, ETH Zurich, 2012.

[14] M. Carandini and D. J. Heeger, "Normalization as a canonical neural computation." *Nature Reviews Neuroscience*, Nov. 2011.

[15] G. Turrigiano and S. Nelson, "Homeostatic plasticity in the developing nervous system," *Nature Reviews Neuroscience*, vol. 5, pp. 97–107, 2004.

[16] A. Siegert, "Phys. Rev. 81, 617 (1951): On the First Passage Time Probability Problem," *Physical Review*, 1951.

[17] L. Ricciardi, *Diffusion processes and related topics in biology*. Springer-Verlag, 1977.

[18] S. Ostojic and N. Brunel, "From spiking neuron models to linear-nonlinear models." *PLoS Computational Biology*, vol. 7, no. 1, p. e1001056, 2011.

[19] W. Gerstner and W. M. Kistler, *Spiking Neuron Models: Single Neurons, Populations, Plasticity*. Cambridge University Press, 2002. [Online]. Available: http://icwww.epfl.ch/ gerstner//BUCH.html

[20] M. Fuenzalida, D. Fernández de Sevilla, A. Couve, and W. Buño, "Role of AMPA and NMDA receptors and back-propagating action potentials in spike timing-dependent plasticity." *Journal of Neurophysiology*, vol. 103, no. 1, pp. 47–54, Jan. 2010.

[21] J. Waters, A. Schaefer, and B. Sakmann, "Backpropagating action potentials in neurones: measurement, mechanisms and potential functions," *Progress in Biophysics and Molecular Biology*, vol. 87, no. 1, pp. 145–170, Jan. 2005.

[22] E. M. Izhikevich and N. S. Desai, "Relating STDP to BCM." *Neural Computation*, vol. 15, no. 7, pp. 1511–1523, Jul. 2003.

[23] Z. Gil and Y. Amitai, "Evidence for proportional synaptic scaling in neocortex of intact animals," *Neuroreport*, vol. 11, no. 18, pp. 4027–4031, Dec. 2000.

[24] G. G. Turrigiano and S. B. Nelson, "Hebb and homeostasis in neuronal plasticity," *Current Opinion in Neurobiology*, vol. 10, no. 3, pp. 358–364, Jun. 2000.

[25] M. Cook, F. Jug, and C. Krautz, "Sharpening projections," *BMC Neuroscience*, Jan. 2009.

[26] ——, "Neuronal Projections Can Be Sharpened by a Biologically Plausible Learning Mechanism," *Artificial Neural Networks–ICANN 2011*, 2011.

[27] R. Douglas, K. Martin, and D. Whitteridge, "A Canonical Microcircuit for Neocortex," *Neural Computation*, Jan. 1989.

[28] R. Douglas and K. Martin, "Mapping the matrix: the ways of neocortex," *Neuron*, Jan. 2007.